

DETAIL



```

graph TD
    Input[外部からの問い合わせ] --> UI[ユーザインタフェース]
    UI --> DB[データベース]
    DB --> M[管理モジュール]
    M --> Output[端末 / プリンタ]
    
```

00/06/26 18:28

[Date of request for examination]
[Date of sending the examiner's decision of rejection]
[Kind of final disposal of application other than the
examiner's decision of rejection or application converted
registration]
[Date of final disposal for application]
[Patent number]
[Date of registration]
[Number of appeal against examiner's decision of
rejection]
[Date of requesting appeal against examiner's decision of
rejection]
[Date of extinction of right]

Copyright (C); 1998 Japanese Patent Office

[MENU](#)

[SEARCH](#)

[INDEX](#)

[DETAIL](#)

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開平8-137697

(43) 公開日 平成8年(1996)5月31日

(51) Int.Cl. ⁶	識別記号	庁内整理番号	F I	技術表示箇所
G 0 6 F 9/44	5 5 2	7737-5B		
9/46	3 6 0 C	7737-5B		
15/16	4 3 0 B			

審査請求 未請求 請求項の数 6 O L (全 12 頁)

(21) 出願番号 特願平6-280262

(22) 出願日 平成6年(1994)11月15日

(71) 出願人 000001199

株式会社神戸製鋼所

兵庫県神戸市中央区脇浜町1丁目3番18号

(72) 発明者 高橋 哲也

兵庫県神戸市西区高塚台1丁目5番5号

株式会社神戸製鋼所神戸総合技術研究所内

(72) 発明者 田村 直樹

兵庫県神戸市西区高塚台1丁目5番5号

株式会社神戸製鋼所神戸総合技術研究所内

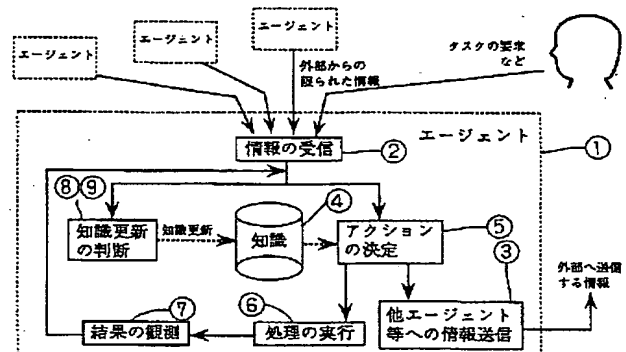
(74) 代理人 弁理士 本庄 武男

(54) 【発明の名称】 自律分散型システム

(57) 【要約】

【目的】 システム設定の変更などの保守の手間を発生することなく、構成要素の変化などによるシステム特性の変化に対応してシステム全体の機能や性能を保ち得る自律分散型システム。

【構成】 本システムは、エージェント①が通信②③により協調動作してタスクを処理する自律分散型システムであって、各エージェント①は自分と他のエージェントのタスク処理能力に関する知識④を持ち、エージェント①同士はそれぞれが持つ知識④に応じて自律的にネゴシーションを行うことによりタスク処理の分担を決定⑤し、各エージェント①はこの決定された分担に従ってタスク処理を実行⑥し、処理結果を観測⑦するに際し、各エージェント①は上記ネゴシーション時の他のエージェントとの情報交換②③及び上記観測⑦結果からシステム特性の変化を推定⑧し、各エージェント①は上記推定されたシステム特性の変化に基づいて自分の持つ知識を逐次更新⑨するように構成されている。上記構成により、システム設定の変更などの保守の手間が不要となる。



【特許請求の範囲】

【請求項 1】 コンピュータシステム内において自律的に動作し、各種の判断や作業を人間に代って行うサブシステムである複数のエージェントが通信により所定のシステム特性に従って協調動作してタスクを処理する自律分散型システムであって、各エージェントは自分と他のエージェントのタスク処理能力に関する知識を持ち、エージェント同士はそれぞれが持つ知識に応じて自律的にネゴシーションを行うことによりタスク処理の分担を決定し、各エージェントは上記決定された分担に従ってタスク処理を実行し、処理結果を観測する自律分散型システムにおいて、各エージェントは他のエージェントとの情報交換及び上記観測結果からシステム特性の変化を推定し、各エージェントは上記推定されたシステム特性の変化に基づいて自分の持つ知識を逐次更新することを特徴とする自律分散型システム。

【請求項 2】 各エージェントは自分と他のエージェントのタスク処理能力を表わす数値パラメータの集合を上記知識として持ち、各エージェントは上記数値パラメータを用いて自分と他のエージェントによるタスク処理コストを推定し上記情報交換に用いる請求項 1 記載の自律分散型システム。

【請求項 3】 各エージェントは上記情報交換により他のエージェントから送られてくる該他のエージェントによるタスク処理コストの推定値及び上記観測結果と自分によるタスク処理コストの推定値とを比較することにより上記システム特性の変化を推定する請求項 2 記載の自律分散型システム。

【請求項 4】 各エージェントは上記タスク処理コストの比較結果に基づいて上記数値パラメータを変化させることにより自分の持つ知識を逐次更新する請求項 3 記載の自律分散型システム。

【請求項 5】 エージェント同士は、上記更新された知識を用いて上記ネゴシーションを行うことにより、各エージェントによるタスク処理コストが最小になるように上記タスク処理の分担を決定する請求項 4 記載の自律分散型システム。

【請求項 6】 各エージェントは、上記知識として他のエージェントにおけるタスク処理の負荷状況を持ち、該知識に基づいて各エージェントの負荷を均等にするように上記ネゴシーションを行う請求項 1 記載の自律分散型システム。

【発明の詳細な説明】

【0001】

【産業上の利用分野】 本発明は、自己分散型システムに係り、詳しくは複数のエージェントが通信により協調動作してタスクを処理する自律分散型システムに関するものである。

【0002】

【従来の技術】 近年、コンピュータネットワークの普及

ならびに計算機のダウンサイジング化が進むにつれて、クライアントサーバ型システムに代表されるようなネットワークによって複数計算機同士を連係作動させるシステムが一般的となってきた。これによって、ネットワークで接続された計算機間の通信を利用して複数のユーザが共同作業を行い、個人の作業効率だけでなくグループとして作業生産性を高めるためのグループウェアやコンカレントエンジニアリングと呼ばれる分野の研究なども盛んとなってきた。また、ディスクや CPU などの資源を共有化して計算機を有効活用することが可能となり、ネットワークやシステムの構成もフレキシブルに行うことが可能となってきた。しかしながら、そのようなシステムが大規模になりネットワークに接続される機器が増加するにつれて、ネットワークシステムを運営していくうえでの問題点も次第に明白になりつつある。すなわち、ネットワークに大量の資源が接続されると数多くのサブシステムが複数計算機に散在することになり、かつてのホスト計算機による集中管理システムとは異なった、システムの管理及び保守の面での困難が生じてくる。例えば、システムを構成する機器の一部に故障や異常が生じた際には、サブシステム同士が通信によって密接に相互依存しながら動作しているために、故障箇所や異常原因を特定することが困難になる。また、サブシステムの数が多い場合にはフレキシブルであったシステム構成も、ネットワークに接続されるサブシステム数がある程度以上増えてしまうと、一部のシステム構成を変更することによる他のサブシステムへの影響を予測することが困難となり、結局はシステムとしてのフレキシビリティが失われてしまう。そこで、最近では「コンピュータシステム内において自律的に動作し、各種の判断や作業を人間に代って行うサブシステム」という意味で使われる「エージェント」という新しい概念を用いたシステムが開発された（情報処理学会研究報告 Vol. 93, No. 69 参照）。これは、複数のエージェントが通信により所定のシステム特性に従って協調動作してタスクを処理する自律分散型システムであって、各エージェントは自分と他のエージェントのタスク処理能力に関する知識を持ち、エージェント同士はそれぞれが持つ知識に応じて自律的にネゴシーションを行うことによりタスク処理の分担を決定し、各エージェントはこの決定された分担に従ってタスク処理を実行し、タスク処理結果を観測するシステムである。ここでは、エージェントはコンピュータのユーザに対して秘書の役割をするソフトウェアや、コンピュータネットワークを利用してシステムの異常監視や診断を行うプロセスなど、様々な役割を担って人間の代行をする。

【0003】

【発明が解決しようとする課題】 上記したような従来の自律分散型システムでは、自分がとるべきアクションを決めるためのルールは、エージェントが持つ知識に基づ

く。この知識には、他のエージェントの特性などの外界に関する知識も含まれるし、自分自身の特性に関する知識も含まれている。しかし、エージェントは必ずしも正しい知識を持つとは限らず、例えばシステム構成の変更などの外界変化や自分自身の特性変動などによってシステム特性が変化すると、エージェントが持つ知識は実際とは異なる誤ったものとなりうる。従って、これら構成要素の変化などによるシステム特性の変化に対応してシステム全体の機能や性能を保つためには、システム設定の変更などの保守の手間をかけざるを得ない。本発明は、このような従来の技術における課題を解決するために、自律分散型システムを改良し、各エージェントが自律的に外界や自分自身の変化を検知して適応動作することにより、システム設定の変更などの保守の手間をかけることなく、システム全体の機能や性能を保持することのできる自律分散型システムを提供することを目的とするものである。

【0004】

【課題を解決するための手段】 上記目的を達成するために本発明は、コンピュータシステム内において自律的に動作し、各種の判断や作業を人間に代って行うサブシステムである複数のエージェントが通信により所定のシステム特性に従って協調動作してタスクを処理する自律分散型システムであって、各エージェントは自分と他のエージェントのタスク処理能力に関する知識を持ち、エージェント同士はそれぞれが持つ知識に応じて自律的にネゴシーションを行うことによりタスク処理の分担を決定し、各エージェントは上記決定された分担に従ってタスク処理を実行し、処理結果を観測する自律分散型システムにおいて、各エージェントは他のエージェントとの情報交換及び上記観測結果からシステム特性の変化を推定し、各エージェントは上記推定されたシステム特性の変化に基づいて自分の持つ知識を逐次更新することを特徴とする自律分散型システムとして構成されている。さらには、各エージェントは自分と他のエージェントのタスク処理能力を表わす数値パラメータの集合を上記知識として持ち、各エージェントは上記数値パラメータを用いて自分と他のエージェントによるタスク処理コストを推定し上記情報交換に用いる自律分散型システムである。さらには、各エージェントは上記情報交換により他のエージェントから送られてくる該他のエージェントによるタスク処理コストの推定値及び上記観測結果と自分によるタスク処理コストの推定値とを比較することにより上記システム特性の変化を推定する自律分散型システムである。さらには、各エージェントは上記タスク処理コストの比較結果に基づいて上記数値パラメータを変化させることにより自分の持つ知識を逐次更新する自律分散型システムである。さらには、エージェント同士は、上記更新された知識を用いて上記ネゴシーションを行うことにより、各エージェントによるタスク処理コスト

が最小になるように上記タスク処理の分担を決定する自律分散型システムである。さらには、各エージェントは、上記知識として他のエージェントにおけるタスク処理の負荷状況を持ち、該知識に基づいて各エージェントの負荷を均等にするように上記ネゴシーションを行う自律分散型システムである。

【0005】

【作用】 本発明によれば、コンピュータシステム内において自律的に動作し、各種の判断や作業を人間に代って行うサブシステムである複数のエージェントが通信により所定のシステム特性に従って協調動作してタスクを処理するに当たり、各エージェントには自分と他のエージェントのタスク処理能力に関する知識を持たされる。エージェント同士によりそれぞれ持たされた知識に応じて自律的にネゴシーションがなされることによって、タスク処理の分担が決定される。各エージェントにより上記決定された分担に従ってタスク処理が実行され、タスク処理結果が観測される。この際、各エージェントにより他のエージェントとの情報交換及び上記観測結果からシステム特性の変化が推定される。そして各エージェントにおいて上記推定されたシステム特性の変化に基づいて自分の知識が逐次更新される。これにより、エージェントが自律的に変化に適応するため、システム管理者などが変化を監視してシステムをチューニングするなどの、人間がシステムを保守する手間を省くことが可能となる。

【0006】

【実施例】 以下添付図面を参照して、本発明を具体化した実施例につき説明し、本発明の理解に供する。尚、以下の実施例は、本発明を具体化した一例であって、本発明の技術的範囲を限定する性格のものではない。ここに、図1は本発明の一実施例に係る自律分散型システムの概略構成を示す模式図、図2はシステム動作の概要を示す模式図、図3は実際の自律分散型システムの概略構成を示す模式図、図4は情報交換の様子を示す説明図、図5は各エージェントが行う判断のルールを示すフロー図、図6は動作結果を示す図表、図7は適応の結果を示す図表、図8は上記自律分散型システムの応用例を示す模式図である。図1に示すごとく、本実施例に係る自律分散型システムは、コンピュータシステム内において自律的に動作し、各種の判断や作業を人間に代って行うサブシステムである複数のエージェント①が通信②③により所定のシステム特性に従って協調動作してタスクを処理する自律分散型システムであって、各エージェント①は自分と他のエージェントのタスク処理能力に関する知識④を持ち、エージェント①同士はそれぞれが持つ知識④に応じて自律的にネゴシーション②③を行うことによりタスク処理の分担等のアクションを決定⑤し、各エージェントはこの決定されたアクションに従ってタスク処理を実行⑥し、タスク処理結果を観測⑦するシステム

である点で従来例と同様である。しかし、本実施例では、各エージェント①は他のエージェントとの情報交換②③及び上記観測⑦結果からシステム特性の変化を推定⑧し、各エージェントは上記推定されたシステム特性の変化に基づいて自分の持つ知識④を逐次更新⑨する点で従来例と異なる。さらに、各エージェント①は自分と他のエージェントのタスク処理能力を表す数値パラメータの集合を知識④として持ち、各エージェント①は上記数値パラメータを用いて自分と他のエージェントとによるタスク処理コストを推定し、上記情報交換②③に用いることとしてもよい。さらに、各エージェント①は上記情報交換②③により他のエージェントから送られてくる該他のエージェントによるタスク処理コストの推定値及び上記観測⑦結果と、自分によるタスク処理コストの推定値とを比較することにより、上記システム特性の変化を推定⑧することとしてもよい。さらに、各エージェント①は上記タスク処理コストの比較結果に基づいて上記数値パラメータを変化させることにより自分の持つ知識④を逐次更新⑨してもよい。さらに、エージェント①同士は上記更新された知識④を用いて上記ネゴシエーションを行うことにより、各エージェント①によるタスク処理コストが最小になるように上記タスク処理の分担を決定⑤するようにしてもよい。

【0007】以下、このシステムの動作について説明する。即ち、図1に示すごとく、エージェント①は外界、即ち他のエージェントとの間で通信②③によって情報交換やネゴシエーションを行う。この時得られた情報を基に、各エージェント①はそれぞれが持つ知識④を用いて自律的にタスク処理の分担決定等の判断を行い、次に自分が取るべきアクションを決定⑤する。ここでいうアクションには大きく分けて2種類ある。一つはエージェント①自身が自分で何らかのタスクを行うことであり、もう一つは他のエージェントと通信②③を行い情報のやり取りをするという動作である。このようなアクションを決めるためのルールは、エージェント①が持つ知識④に基づく。知識④には、他のエージェントの特性などの外界に関する知識も含まれるし、自分自身の特性に関する知識も含まれている。エージェントは必ずしも正しい知識を持つとは限らず、例えばシステム構成の変更などの外界変化や自分自身の特性変動などによってシステム特性が変化すると、エージェントが持つ知識は実際とは異なる誤ったものとなり得る。この点については従来例にて述べた通りである。従って、エージェント①が行う判断を常に適正なものに保持するには、エージェント①が外界や自分自身の変化を検出して適応する機能が有効である。すなわち、エージェント①が自律的に変化に適応する機能を持つようにすることにより、システム管理者などが変化を監視してシステムをチューニングするなどの、人間がシステムを保守する手間を省くことが可能となる。このように適応を行うために、自分以外の外界の

変化をエージェント①が得る手段は、基本的には他のエージェントとの通信②③による情報の授受である。一方、自分自身の特性変化などを検出する手段としては、エージェント①が自らタスクを行った際には、エージェント①はなんらかの実行経過を観測⑦するものとし、その観測⑦結果から自らの特性を推定⑧する機能を持つようにすることとした。以下、エージェント①の概念をさらに具体化するために、簡単な自律分散型システムを想定した複数エージェントによるシステムのシミュレータを例にとって説明する。

【0008】本シミュレータでは、図2に示すように、複数のサーバとなるプロセスがコンピュータ内にエージェントとして存在する状況を想定し、不規則に発生する種々のタスクの割り当てを決定する問題を扱う。つまり、クライアントから出されるタスクの実行要求に対して、各タスクをどのエージェントが処理するのかを、エージェント同士がネゴシエーションしながら決定する。従って、ここでは最適な仕事の割り当てを行うようにコントロールを行うスーパーバイザは存在せず、あくまでもエージェントの自律的な話し合いつまりネゴシエーションによって役割分担の決定がなされる。実際の自律分散型システムでは、図3に示すように、個々のエージェントは独立したプロセスとして実行され、エージェント同士の情報のやりとりはプロセス間通信によって行われる。この仕組みでは、各エージェントは異なる計算機上で動作させられることを可能としており、その際にはエージェント間の通信はネットワークを介した通信にて行われる。このため、各エージェントはそれぞれの通信バッファを有しており、他のエージェントから送られてきた情報は一旦通信バッファ内のキューに蓄積される。従って、そのシミュレータにおけるエージェント間の通信はエージェント同士のリアルタイムの対話ではなく、むしろメール型の通信方式であるといえる。つまり、エージェントが送信した情報は相手の通信バッファに届けられ、相手のエージェントは都合のよいタイミングで適宜その情報を参照して自らの判断を行う。情報を送信したエージェントもその時点では相手からの返事を待つことはせず、送信後はすぐに他の動作を開始できる。このような通信方式にすることにより、エージェントが通信によって拘束されることがなくなり自律性がさらに高まると考えられる。

【0009】各エージェントは、自分も含めた全てのエージェントの特性に関する知識を持っている。ここにいうエージェントの特性は、例えばエージェントの持つ処理スピードや使える記憶領域のサイズなどといった、エージェントがタスクを実行する際の性能に影響を及ぼすものを想定している。各エージェントはタスク実行の要求を受信した際には、その知識に基づいて自分でタスクを実行すべきか、あるいは他のエージェントに依頼すべきかを判断する。一方、依頼を受けたエージェント側で

も、その依頼を受けるべきか、あるいは拒否すべきかを自分の知識を用いて判断し、返答を依頼元に送信する。先述したように、エージェントが持つ知識は必ずしも正しいとは限らず、知識が誤っている場合もあれば、環境変化などにより現実とのずれが発生することもあり得る。そこで、各エージェントは通信内容や観測結果から知識の誤りを推定し、自らの知識を更新して適応していく。具体的には、エージェント間で行われるネゴシエーションの過程で互いの持つ情報の一部が交換され、エージェントは他から得た情報をもとに自らの知識を更新する。これにより、エージェントは外界の変化などに対して適応していくことになる。また、エージェントは自らタスク実行などを行った際には、その結果を観測することにより自分自身に関する知識と現実とのずれを調べ、自分についての知識も更新する。このような適応の機能により、エージェントは動作を常に最適に保持しようとする。引き続いて、本シミュレータにおける具体的な処理（シミュレーション）の内容を説明する。

【0010】 先ず、システムに対するタスクの要求は、システムの外部よりいずれかのエージェントに対してなされる。これは、例えばコンピュータのユーザがシステムに対して「データベース内から、あるデータを取り出せ」という命令を発した際に、最初にひとつのエージェントがそれをタスク要求として受け取ることに相当する。ここではタスクは、各エージェントに対してランダムに発生させるようにした。具体的にはM個のエージェント $m=1, 2, \dots, M$ に対して、要求を受け取る確立 p_m ($\sum_m p_m = 1$) を設定しておき、シミュレーション中では乱数発生によってタスク要求を設定した割合でエージェントに振り分けた。本シミュレータではタスク自身の発生頻度は一定間隔でもシミュレーションできるが、待行列理論により一般的に用いられているポアソン分布のモデルに従っても行える。ここで各タスクには、パラメータとして「要因」 $a = [a_1, a_2, \dots, a_n]$ を設定する。要因の中に含まれる各パラメータは数値パラメータであって、例えば「CPU能力が多く必要」とか「記憶容量が多く必要」といった、タスクの特性を数値で表すものである。これらの要因の中の各パラメータは正規分布によってランダムに決定し、それぞれのタスクは異なる特性を持ったものとして発生させる。本シミュレーションでは、エージェントは与えられたタスクの要因値を知ることができるものとし、その値をもとにした判断により互いにネゴシエーションして仕事の分担を決定することとした。

【0011】 次に、エージェントの能力を表すパラメータとして、「スキル」 $s = [s_1, s_2, \dots, s_n]$ を設けた。スキルの中のパラメータ s_n は、これも数値パラメータであって、それぞれの要因パラメータ a_n に対応し、タスクに含まれる各要因に対するエージェントが持つ処理能力を表す。例えば、各スキルパラメータは要

因パラメータに対応して「エージェントのCPUスピード」であるとか、「エージェントが使用できる記憶容量」などを表す値である。エージェントがタスクを処理するのに要するコストは、要因とスキルとの組合せによって関数 $f(a, s)$ で表現する。この場合、エージェントのスキルに対してタスクの要因値が大きい場合にコストが大きくなるように関数の形を決めておく必要がある。本シミュレーションでは次式のように定めた。

$$f(a, s) = A \cdot \sum (\exp(a_n - s_n))$$

ここで、Aは正の定数である。後述するように、このコストはエージェントが本シミュレーション内でタスクを処理する処理時間としても用いる。次に、エージェントは、各エージェント1, 2, ..., Mのスキルに関する知識 s_1, s_2, \dots, s_M を持つものとする。タスク処理の要求を受けた際に、その知識に基づいて他のエージェントが自分より低コストで処理できると予想した場合には、エージェントはタスクを他のエージェントに依頼する。この依頼を受けたエージェントも自らの知識により、依頼元の方が低コストであると予想すると依頼を拒否する。このようなネゴシエーションの過程において、エージェント同士は互いが持つ情報の一部を相手に伝達する。本シミュレーションでは、各々が判断材料とした予想コストを相手と交換することとした。つまり、他のエージェントへ依頼を行う際、並びに依頼に対して承諾や拒否を行う際には、図4に示すように、自分及び相手が対象タスクを処理した場合のコストの予想値を同時に送信する。

【0012】 図5に、本シミュレーションにて行ったネゴシエーションのルールのフロー図を示した。このフロー図から分かるように、エージェントは依頼が拒否されても、自分より低コストで処理できそうな相手が存在する場合には、次々と依頼を繰り返す。また、依頼を承諾したエージェントも、より低コストとなりそうな相手を見つければ、そのエージェントに再依頼する。この場合、同じエージェントに対して何回も同じタスクの依頼がされないように、ネゴシエーションの過程ではタスクの要因パラメータと一緒に依頼の履歴も送信するようにした。以上のように、本シミュレーションでエージェントが知識更新を行うために、外部から得ることができる情報は、ネゴシエーション時に受信する他のエージェントによる予想コストだけである。従って、外部から得られる情報は相手エージェントの知識から加工されたデータのみであり、その元となる相手の知識は必ずしも正しいとは限らない。そこで、各エージェントは得られた情報から実世界を推定する必要がある。そこで、本シミュレーションでは、一般によく用いられる適応則に基づいて各エージェントが自分の知識を更新し、知識を真の値に収束させることを試みた。以下、この適応規則について述べる。まず、あるエージェントが、要因 a を持つタスクに関するネゴシエーションを、他のエージェント m

と行う場合を考える。その際には、エージェントは自分の持つ他のエージェントmのスキルに関する知識 s_m を用いて、他のエージェントmがそのタスクを行ったときに生じるコスト y を $y = f(a, s_m)$ により予測する。

【0013】一方、ネゴシエーションの際には、相手即ち、他のエージェントmからのm自身が自分で予想した処理コストが送られてくる。この送られてきた他のエージェントm自身による予想コストを y として、他のエージェントmのスキルに関する知識は次式に示すような形で更新される。

【数1】

$$\Delta s_m = K(a, s_m) \cdot (y - \hat{y}) \cdot \frac{\partial f}{\partial s} \Big|_{s=s_m}$$

すなわち、他のエージェントm自身が予想した値 y の信頼性が高いと判断し、予測値が y に近づくように知識 s_m を更新する。ここで、 $K(\dots)$ は適応のゲインを調整する関数である。以上は、エージェントが自分の外界に関する知識を適応させるための処理であったが、エージェントが持っている自分自身のスキルに関する知識も常に正しいとは限らないので、自分に関する知識も更新する必要がある。そのために、エージェントは自分自身がタスクを処理した場合には、処理にかかったコストを観測する。この場合、観測した処理コストを y とし、自分を m とすれば、全く上記と同様に知識の更新が行える。

【0014】引き続き本シミュレーションの結果について述べる。ここでは要因およびスキルが3パラメータからなり、3個のエージェントで構成されるシステムのシミュレーションを行った。前述したように、タスクの要求は全てのエージェントに対してランダムになされ、エージェント同士はネゴシエーションによってタスクの分担を決定する。担当となったエージェントはタスクを処理するが、その際にコストに相当する時間を処理に要する。従って、コストの大きなタスクを行うと長い処理時間を要することになり、そのエージェントが処理すべき他のタスクが存在する場合には長く待たされる結果にもなる。従って、前述したようなモデルでは、理論上は無限に長い処理時間が発生する可能性があるが、それではシミュレーションをするうえで不都合である。そこで、処理時間が一定値以上になった場合にはタイムアウトとして処理を打ち切ることとした。適応の結果を次に示す。図6には、本シミュレーションでの動作結果の一例として、エージェント1のスキルのパラメータが変化したのに対し、タスク処理やネゴシエーションを繰り返すうちに、各エージェントの知識 s_i が時間とともに正しい値に収束していく様子が示されている。つまり、時刻0以前では、知識は $s_1 = (1.0, 0.0, 0.0, -1.0)$ であり、全てのエージェントが正しい知識を有

している。これに対し、時刻0で知識 s_1 が $(-1.$

$0, 0.0, 1.0)$ に変化した状況を想定する。この変化を各エージェントが検知し、時間とともに知識が正しく推定されていく様子が、図6より分かる。

【0015】次に、図7にはエージェントが適応前の知識を用いたネゴシエーションによりタスクの割り当てを行った場合と、適応後の知識によってタスクの割り当てを行った場合の、各タスクの処理時間の分布が示されている。図7より、適応後のほうがタスクがより短時間で処理できる適切なエージェントに割り当てられる様子が分かる。本シミュレーションでは、適応機能を持ったエージェント群からなるシステム構築の可能性を確認することができた。上記実施例では、エージェントが用いたネゴシエーションのルールは、各タスクを最適なエージェントに割り当てようとするものであった。しかしながら、このルールはローカルな判断のみによって決定を行うものであり、システム全体の性能までは考慮されていない。例えば、前記図7に示した適応の結果では、各タスクの処理時間は短縮していたにもかかわらず、適応前よりも適応後のほうがリードタイム（各タスクが要求されてからネゴシエーションや待ち時間を経て処理完了するまでの時間）は逆に増加していた。これは、適応後の知識を用いたネゴシエーションでは特定のエージェントへのタスクの割り当てが集中したためである。このことは、各エージェントが上記知識として他のエージェントにおけるタスク処理の負荷状況を持ち、この知識に基づいて各エージェントの負荷を均等にするように上記ネゴシエーションを行うことにより解決できる。また、前記図6での適応の過程を見て分かるのとおり、エージェント1の知識の真値への収束が他のエージェントの知識の更新スピードに比較して早く進んでいる。これは、エージェント1のパラメータ変化は、まずエージェント1自身がタスク処理結果の観測によって検知することによるものである。

【0016】一方、他のエージェントの適応過程をみると、適応がスムーズに進む時と、そうでない時とがあるというバラツキ現象が見られる。これは、エージェント1の特性変化に関する情報を他のエージェントが得る手段はネゴシエーション時のコスト値の交換だけであり、知識の誤りを検出できる情報が伝達されるようなネゴシエーションが行われた場合にのみ知識が有効に更新されるからである。従って、知識の誤りが予想コストに顕著に現れるようなタスクが多く発生し、そのような情報をネゴシエーションによって得る機会が多いエージェントの場合には適応が速く進み、逆に情報を得る機会が少ないエージェントでは知識の誤りが是正されにくいということになる。したがって、通信する機会が少なかったり、偏った情報ばかり受信してしまうようなエージェントの発生を防ぐように長期間通信していないエージェントとも定期的に情報交換をするなどの通信ルールとする

ことによりこの不具合をなくすることができる。

【0017】以上のように、本実施例では、複数エージェントが通信によってネゴシエーションを行い、通信によって得た知識からシステム特性変化などを検知して自律的に適応し、ローカルな知識を更新していく新しい仕組みを持った情報システムを実現できた。その結果、各サブシステムが自律的に外界や自分自身の変化を検知して適応動作することにより、システム設定の変更などの保守の手間を発生することなく、構成要素の変化などによるシステム特性の変化に対応してシステム全体の機能や性能を保つことができる自律分散型システムを得ることができた。このようなエージェントによって構成されたシステムが動作するイメージを、図8に、分散データベースの検索処理並びにF A工場における物流制御を想定して描いたものを呈示した。尚、トータル最適化のための各エージェントのルール設計や、適応機能の性能向上のための情報伝達の仕組みや通信プロトコルの設計など、システム性能向上のための設計手法を検討していくことにより、さらにシステムの効率化を計ることができるものと考えられる。

【0018】

【発明の効果】本発明に係る自律分散型システムは、上記したように構成されているため、各サブシステムが自律的に外界や自分自身の変化を検知して適応動作することにより、システム設定の変更などの保守の手間を発生することなく、構成要素の変化などによるシステム特性

の変化に対応してシステム全体の機能や性能を保つことができる。

【図面の簡単な説明】

【図1】 本発明の一実施例に係る自律分散型システムの概略構成を示す模式図。

【図2】 システム動作の概要を示す模式図。

【図3】 実際の自律分散型システムの概略構成を示す模式図。

【図4】 情報交換の様子を示す説明図。

【図5】 各エージェントが行う判断のルールを示すフロー図。

【図6】 動作結果を示す図表。

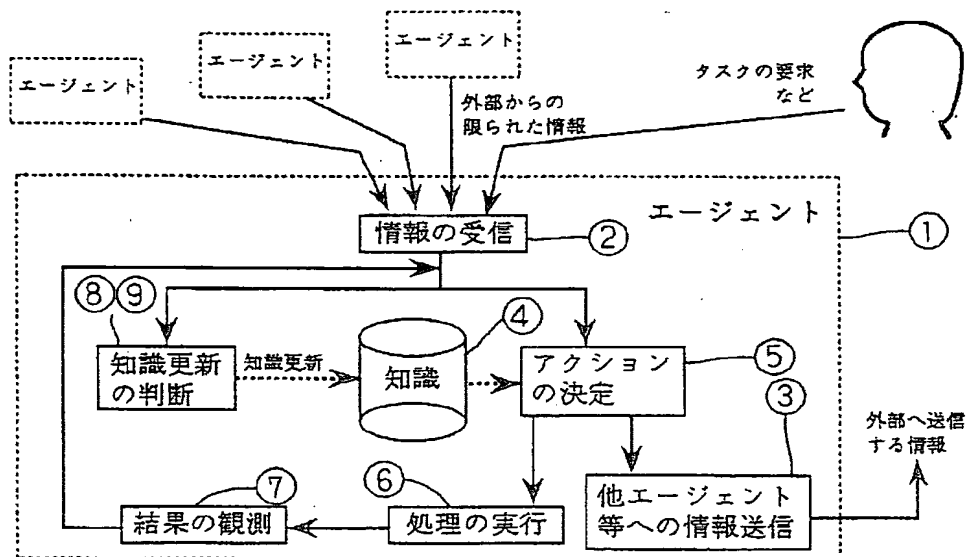
【図7】 適応の結果を示す図表。

【図8】 上記自律分散システムの応用例を示す模式図。

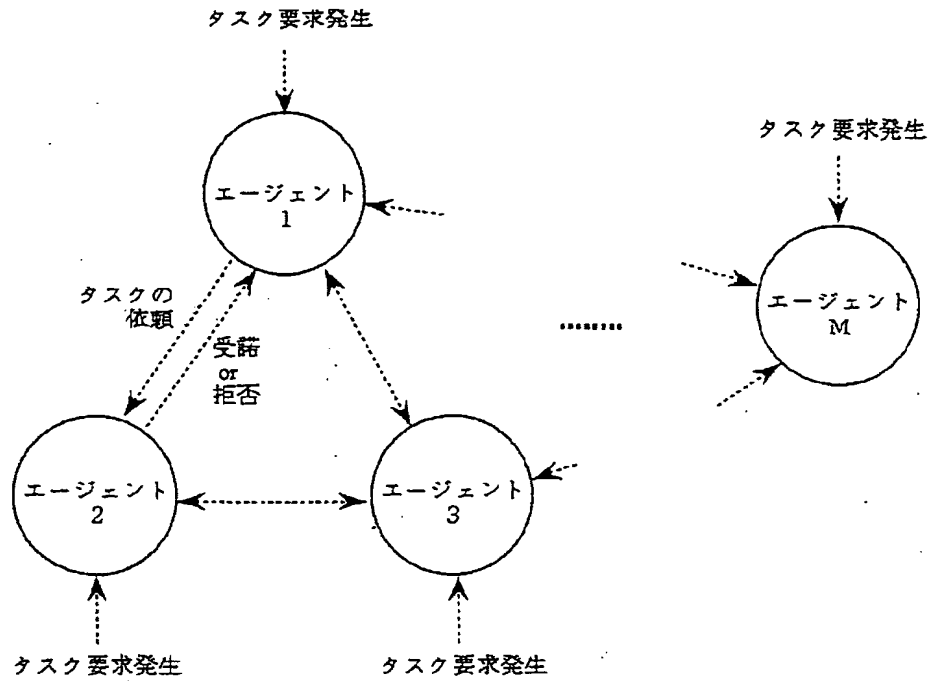
【符号の説明】

- ①…エージェント
- ②…情報の受信
- ③…情報の送信
- ④…知識
- ⑤…アクションの決定（分担決定を含む）
- ⑥…処理の実行
- ⑦…結果の観測
- ⑧…システム特定の変化の推定
- ⑨…知識の更新

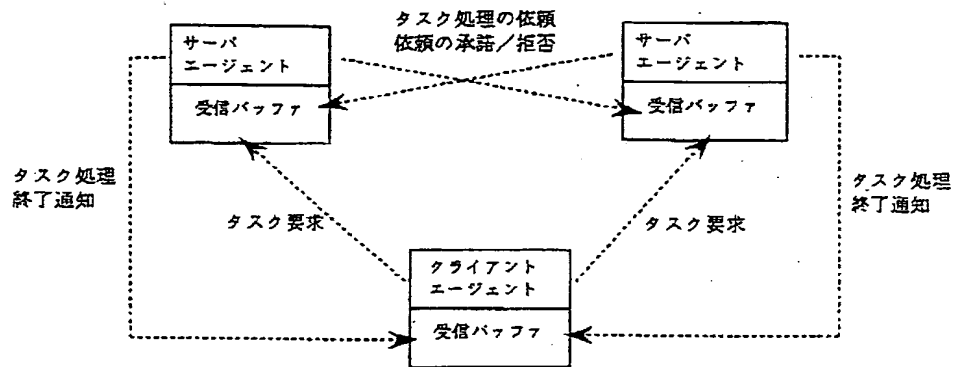
【図1】



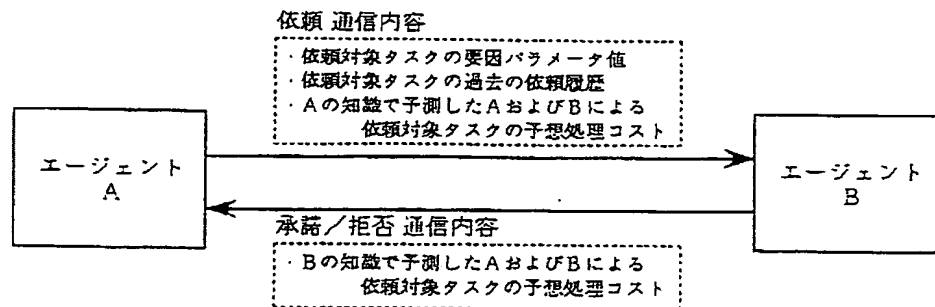
【図2】



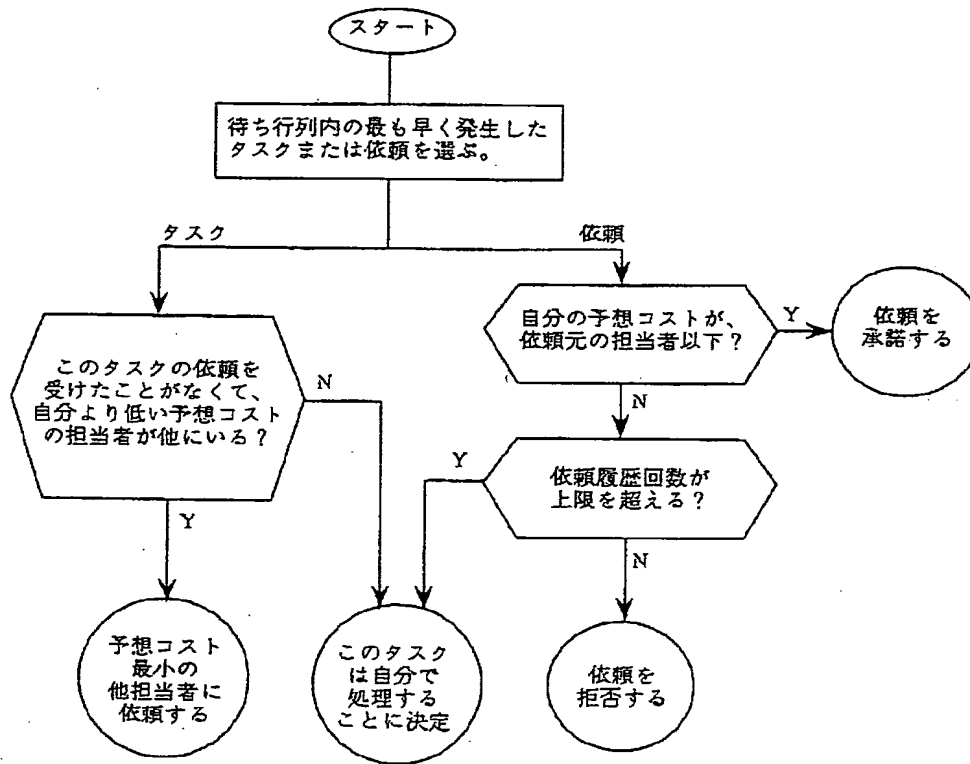
【図3】



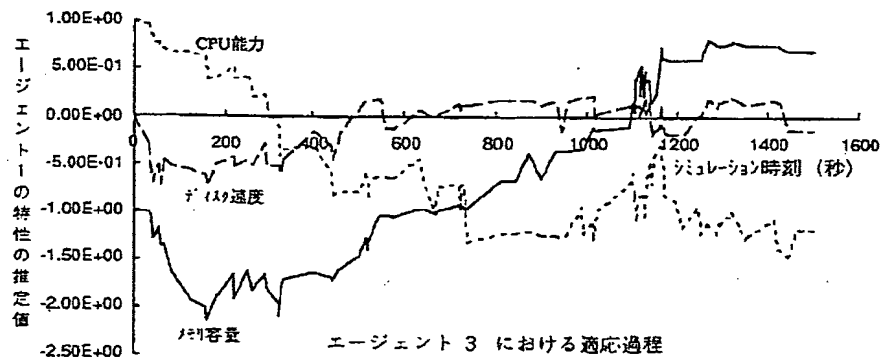
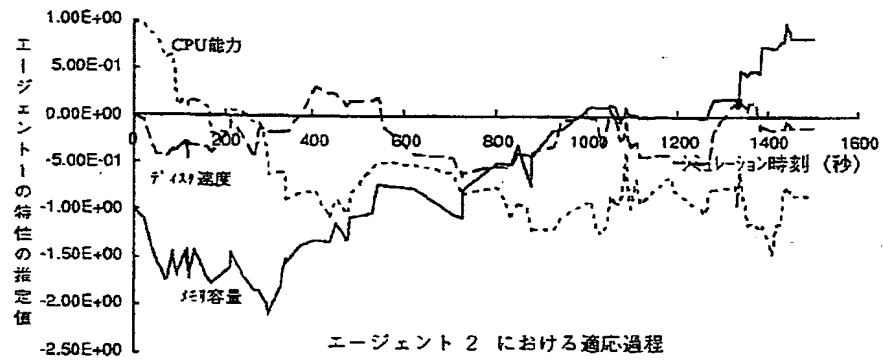
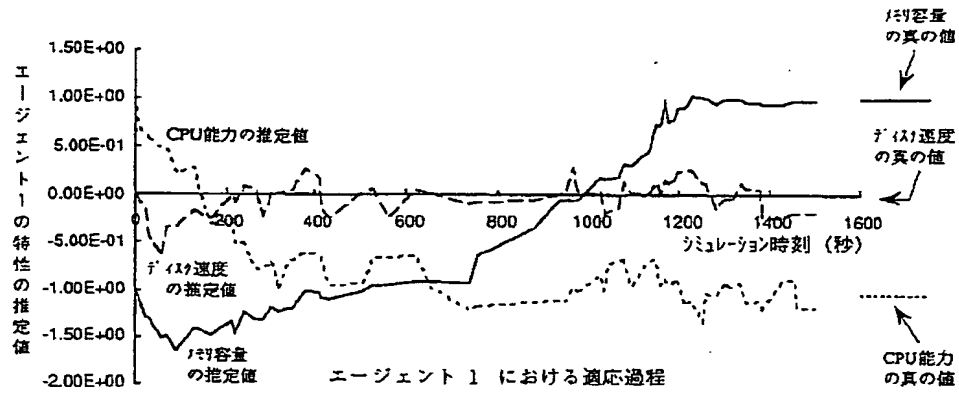
【図4】



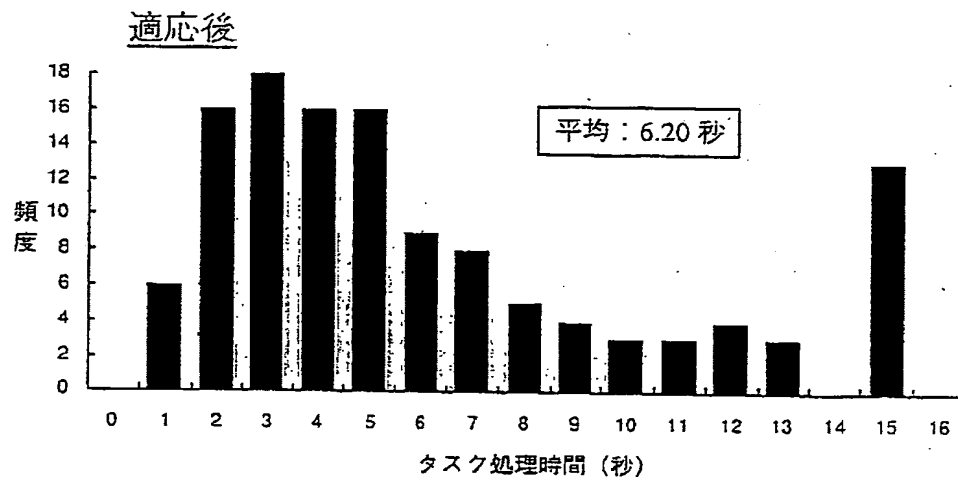
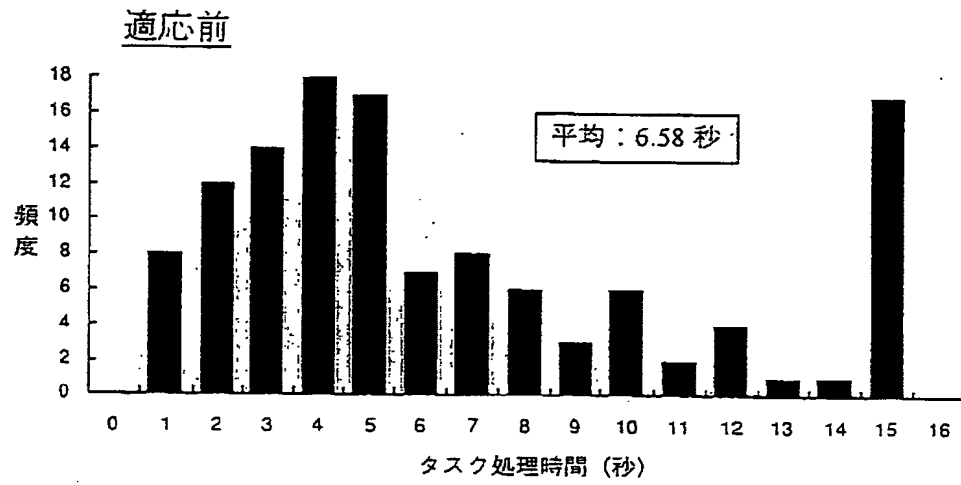
【図 5】



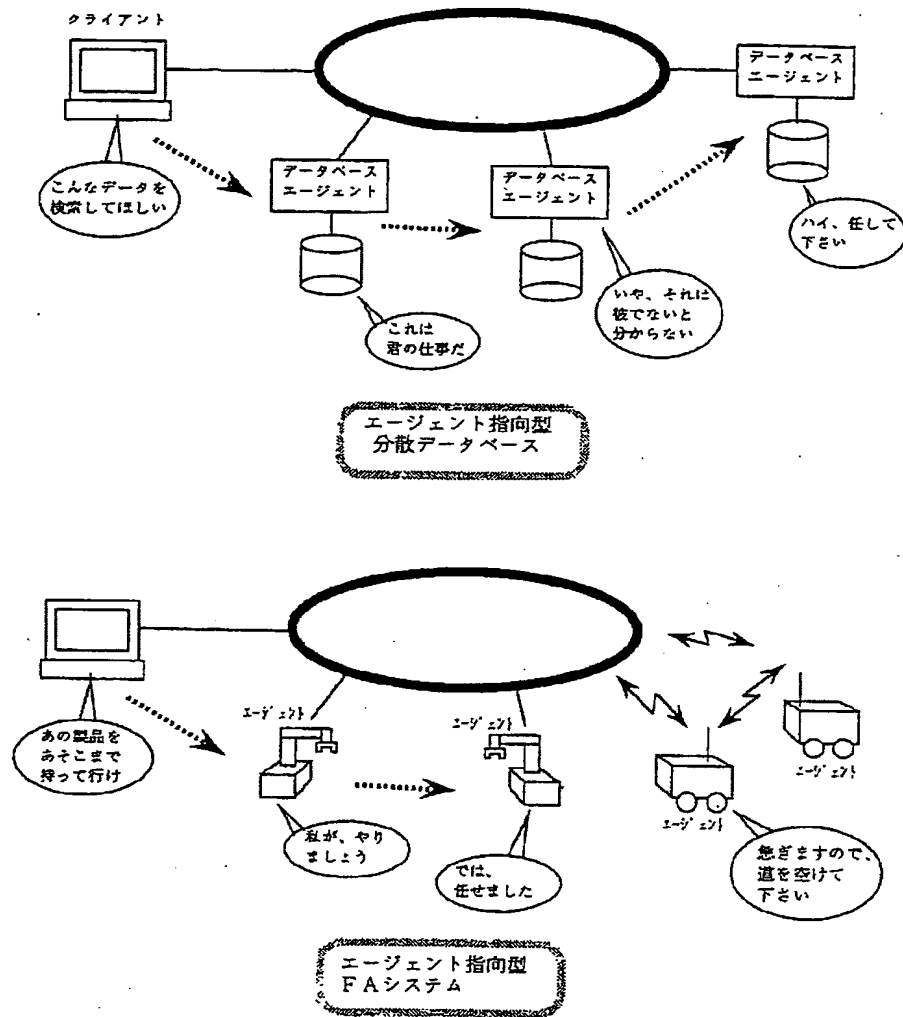
【図6】



【図7】



【図8】



**This Page is Inserted by IFW Indexing and Scanning
Operations and is not part of the Official Record**

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- ☐ **BLACK BORDERS**
- ☐ **IMAGE CUT OFF AT TOP, BOTTOM OR SIDES**
- ☐ **FADED TEXT OR DRAWING**
- ☒ **BLURRED OR ILLEGIBLE TEXT OR DRAWING**
- ☐ **SKEWED/SLANTED IMAGES**
- ☐ **COLOR OR BLACK AND WHITE PHOTOGRAPHS**
- ☐ **GRAY SCALE DOCUMENTS**
- ☐ **LINES OR MARKS ON ORIGINAL DOCUMENT**
- ☐ **REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY**
- ☐ **OTHER:** _____

IMAGES ARE BEST AVAILABLE COPY.

As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.